World Cup

MsCA Sports Analytics

November, 04 2018

Overview

Models will be built to focus on the following tasks:

- Expected goals: all passes that end in a shot
- Expected possession: all possession for a team
- Pass map

Data

All the data for the three models exists in `data/events/19714.json`. `19714` represents one game. The events folder has events for all games. There are corresponding `matches` and `lineups` data sets that tie back to the events. We will focus primarily on `events` for now

```
events_json <- fromJSON("data/events/19725.json", simplifyVector = FALSE)
```

Parse `events_json` to extract relevant data for **expected goals** model.

```
replace_na_empty <- function(x) {
  map_if(x, is.null, ~ NA)
}

event_id <- map_chr(events_json, ~ .x$id)

type_id <- map_int(events_json, ~ .x$type$id)

type_name <- map_chr(events_json, ~ .x$type$name)

timestamp <- map_chr(events_json, ~ .x$timestamp)

possession_team_name <- map_chr(events_json, ~ .x$possession_team$name)

team_name <- map_chr(events_json, ~ .x$team$name)

pass_length <- map(events_json, ~ .x$pass$length) %>%
  replace_na_empty() %>% unlist()

pass_height <- map(events_json, ~ .x$pass$height$name) %>%
  replace_na_empty() %>% unlist()

pass_angle <- map(events_json, ~ .x$pass$angle) %>%
  replace_na_empty() %>% unlist()

duration <- map(events_json, ~ .x$duration) %>%
  replace_na_empty() %>% unlist()
```

```r
play_pattern_name <- map_chr(events_json, ~ .x$play_pattern$name)

goalkeeper_type_name <- map(events_json, ~ .x$goalkeeper$type$name) %>%
  replace_na_empty() %>% unlist()

goalkeeper_outcome_name <- map(events_json, ~ .x$goalkeeper$outcome$name) %>%
  replace_na_empty() %>% unlist()
```

```r
events_df <- data.frame(
    event_id,
    type_id,
    type_name,
    timestamp,
    duration,
    team_name,
    possession_team_name,
    play_pattern_name,
    pass_length,
    pass_height,
    pass_angle,
    goalkeeper_type_name,
    goalkeeper_outcome_name
  ) %>%
  #' Used to identify sequences; max `FALSE` value is the start of a sequence
  mutate(lead_possessor = possession_team_name == lead(possession_team_name)) %>%
  as_tibble()
```

Check index of all shots and then look back to see what lead to a shot.

```r
(shot_indexes <- which(str_detect(events_df$type_name, "Shot")))
```

```
##  [1]  120  466  518  715  904 1082 1624 1663 1699 1775 1848 1988 2476 2481
## [15] 2525 2658
```

Pass sequences are defined as uninterrupted possession leading to a shot.

```r
sequence_indexes <- vector("list", length(shot_indexes))

for (i in seq_along(shot_indexes)) {
  start_index <- ifelse(i == 1, 1, shot_indexes[i - 1] + 2)
  sequence_indexes[[i]] <- seq(start_index, shot_indexes[i] + 1, 1)
}

shots_split <- map(sequence_indexes, ~ events_df %>% slice(min(.x):max(.x)))

start_sequence <- map_int(shots_split, function(x) {
  x <- x %>% mutate(type_name_flag = ifelse(lag(type_name) == "Shot", "remove", "keep"))
  x <- x %>% filter(type_name_flag == "keep")
  #' Check if `FALSE` exist and return max index of `FALSE`
  if (length(which(!x$lead_possessor)) >= 1) {
    as.integer(max(which(!x$lead_possessor)) + 2)
  } else {
      min(x$lead_possessor)
    }
  }
```

```
)

pass_sequences <- map2(shots_split, start_sequence, ~ .x %>% slice(.y:nrow(.x))) %>%
  map2(., 1:length(shots_split), ~ mutate(.x, pass_sequence_label = .y)) %>%
  bind_rows() %>%
  mutate(
    pass_sequence_label = factor(pass_sequence_label),
    type_name = ifelse(type_name == "Goal Keeper", goalkeeper_type_name, type_name)
  ) %>%
  group_by(pass_sequence_label) %>%
  # Need to account for shots that were block by someone
  # other than the goal keeper when identifying outcome of shot
  mutate(goals = case_when(
        str_detect(goalkeeper_outcome_name, "Goal Conceded|Penalty Conceded|No Touch|Touched In") ~ "goa
        str_detect(goalkeeper_type_name, "Shot Faced") & lag(type_name) == "Shot" ~ "missed shot",
        is.na(goalkeeper_outcome_name) & is.na(goalkeeper_type_name) ~ NA_character_,
        lag(type_name) != "Shot" | is.na(lag(type_name)) ~ NA_character_,
        TRUE ~ "saved"
  ))

pass_sequences %>%
  select(pass_sequence_label, type_name, goalkeeper_outcome_name, goals) %>%
  filter(!is.na(goals)) %>%
  distinct(pass_sequence_label, goals)
```

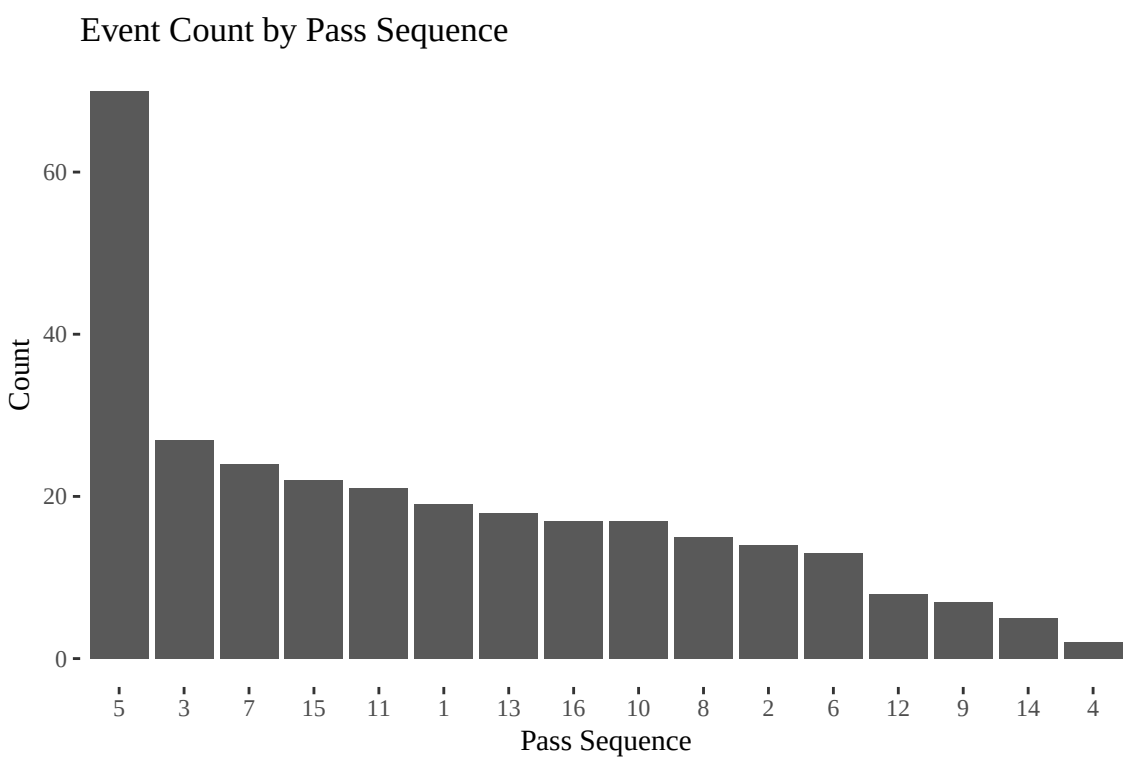
```
## # A tibble: 15 x 2
## # Groups:   pass_sequence_label [15]
##    pass_sequence_label goals
##    <fct>               <chr>
##  1 1                   missed shot
##  2 2                   missed shot
##  3 3                   missed shot
##  4 4                   saved
##  5 5                   missed shot
##  6 6                   missed shot
##  7 7                   missed shot
##  8 8                   saved
##  9 9                   missed shot
## 10 10                  saved
## 11 11                  saved
## 12 12                  missed shot
## 13 14                  goal conceded
## 14 15                  missed shot
## 15 16                  missed shot
```

```
pass_sequences %>%
  group_by(pass_sequence_label) %>%
  slice(1:6) %>%
  select(pass_sequence_label, everything())
```

```
## # A tibble: 91 x 16
## # Groups:   pass_sequence_label [16]
##    pass_sequence_l~ event_id type_id type_name timestamp duration team_name
##    <fct>            <chr>      <int> <chr>     <chr>        <dbl> <chr>
```

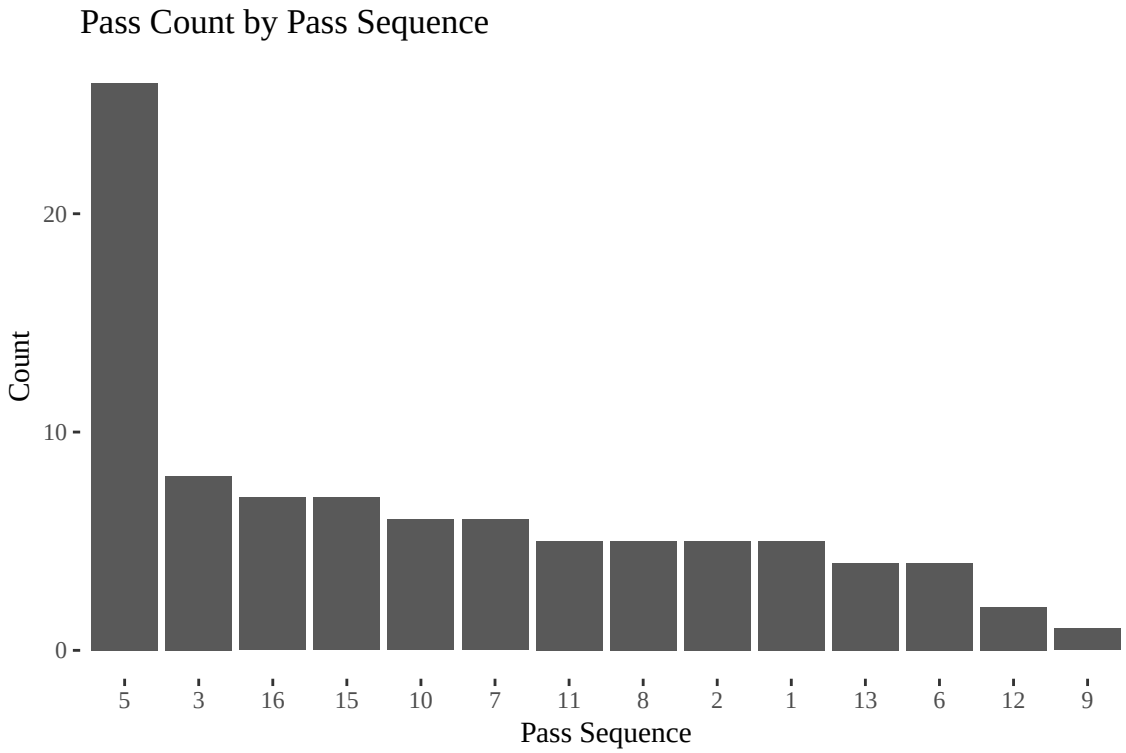
```
##  1 1                a394073~      30 Pass      00:02:23~    3.48  Liverpoo~
##  2 1                bd1a339~      42 Ball Rec~ 00:02:27~   NA     Liverpoo~
##  3 1                10c7de6~      30 Pass      00:02:27~    1.38  Liverpoo~
##  4 1                63a8e0c~      17 Pressure  00:02:27~    0.349 Brighton~
##  5 1                feef5bc~      42 Ball Rec~ 00:02:28~   NA     Liverpoo~
##  6 1                8762a4f~      22 Foul Com~ 00:02:29~    0     Brighton~
##  7 2                5bbf2e2~      30 Pass      00:14:32~    0.649 Liverpoo~
##  8 2                e0383ea~      42 Ball Rec~ 00:14:33~   NA     Liverpoo~
##  9 2                549ec2b~      30 Pass      00:14:34~    1.36  Liverpoo~
## 10 2                306d6c8~      42 Ball Rec~ 00:14:35~   NA     Liverpoo~
## # ... with 81 more rows, and 9 more variables: possession_team_name <chr>,
## #   play_pattern_name <chr>, pass_length <dbl>, pass_height <chr>,
## #   pass_angle <dbl>, goalkeeper_type_name <chr>,
## #   goalkeeper_outcome_name <chr>, lead_possessor <lgl>, goals <chr>
```

```r
pass_sequences %>%
  group_by(pass_sequence_label) %>%
  count() %>%
  ggplot(aes(fct_rev(fct_reorder(pass_sequence_label, n)), n)) +
  geom_col() +
  labs(title = "Event Count by Pass Sequence",
       x = "Pass Sequence",
       y = "Count")
```




```r
pass_sequences %>%
  count(pass_sequence_label, type_name) %>%
  filter(type_name == "Pass") %>%
  ggplot(aes(fct_rev(fct_reorder(pass_sequence_label, n)), n)) +
  geom_col() +
  labs(title = "Pass Count by Pass Sequence",
       x = "Pass Sequence",
```

```
       y = "Count")
```


Pass Count by Pass Sequence
Count
20
10
0
5
3
16
15
10
7
11
8
2
1
13
6
12
9
Pass Sequence